Evolving Electronic Circuits for Computational Intelligence Hardware

Adrian Stoica, R. Zebulum, D. Keymeulen, Y. Jin and T. Daud
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
818-354-2190
adrian.stoica@jpl.nasa.gov

Abstract

This paper discusses the use of Evolvable Hardware (EHW) in automatic synthesis of electronic circuits for computational intelligence (CI) hardware. EHW refers to HW design and self-reconfiguration using evolutionary/genetic mechanisms. Evolutionary experiments in simulations and with a Field Programmable Transistor Array (FPTA) chip in-the-loop demonstrate automatic synthesis of electronic circuits. Unconventional circuit (e.g. implementing CI mechanisms) for which there are no textbook design guidelines are particularly appealing to EHW. To illustrate this situation, one demonstrates here the evolution of circuits implementing parametrical connectives for fuzzy logics.

1. Introduction

The application of evolution-inspired formalisms to hardware design and self-configuration lead to the concept of evolvable hardware (EHW). In the narrow sense, EHW refers to self-reconfiguration of electronic hardware by evolutionary/genetic reconfiguration mechanisms. In a broader sense, EHW refers to various forms of hardware from sensors and antennas to complete evolvable space systems that could adapt to changing environments and, moreover, increase their performance during their operational lifetime.

The paper starts with an overview of the main concepts of EHW. It then describes an effort toward building evolution-oriented devices and an evolvable system on a chip. A Field Programmable Transistor Array architecture is used as the experimental platform for evolutionary experiments. The platform is quite flexible and supports implementation of both analog and digital circuits. While previous works [1:3] illustrated the implementation of several conventional building blocks for electronic circuits such as logical gates, transconductance amplifiers, filters, gaussian neuron, etc., this paper illustrates the automatic design of the rather more unconventional circuits for combinatorial fuzzy logics.

The paper is organized as follows: Section 2 presents the components of an evolvable hardware system. Section 3 surveys some important evolutionary experiments and applications of evolvable hardware. Section 4 presents an evolution-oriented architecture based on the concept of Field Programmable Transistor Array. Section 5 illustrates how the FPTA can be used to evolve reconfigurable circuits for combinatorial fuzzy logic. Circuits implementing parametric triangular norms are evolved in software and in hardware directly on the chip.

2. Evolutionary synthesis of electronics

The main idea of evolutionary/genetic algorithms is inspired by the principle of natural selection. In nature the fittest individuals survive and reproduce passing along their genetic material to their offspring, who will inherit the characteristics that made the parents successful. Similarly, the evolution of artificial systems is based on a population of competing designs, the best ones (i.e. the ones that come closer to meeting the design specifications) being selected for further investigation. The offspring of this elite, in which pairs of parents were randomly selected for "mating", combine genetic material from two parents and may suffer genetic "mutations" (alternatively, in asexual reproduction the genetic code from one successful individual may be inherited, possibly with some random mutation). The offspring are the new generation of competing designs. This process of trial-and-error parallel search can last many generations, and can be constructed with many choices on how to implement reproduction, selection, etc.

The concept of evolvable hardware was born partially inspired by search/optimization/adaptation mechanisms and partially by the availability of reconfigurable

devices such as Field Programmable Gate Arrays (FPGA). Circuits can be evolved reconfiguring programmable devices (which is called *intrinsic* EHW) or evolving software models – descriptions of the electronic HW (referred to as *extrinsic* EHW).

Figure 1 illustrates the main steps of evolutionary design for electronic circuits. Each candidate circuit design is associated with a "genetic code" or chromosome. The simplest representation of a chromosome is a binary string, a succession of 0s and 1s that encode a circuit. The first step of evolutionary synthesis is to generate a random population of chromosomes. The chromosomes are then converted into a model that gets simulated (e.g. by a circuit simulator such as SPICE) and produces responses that are compared against specifications.

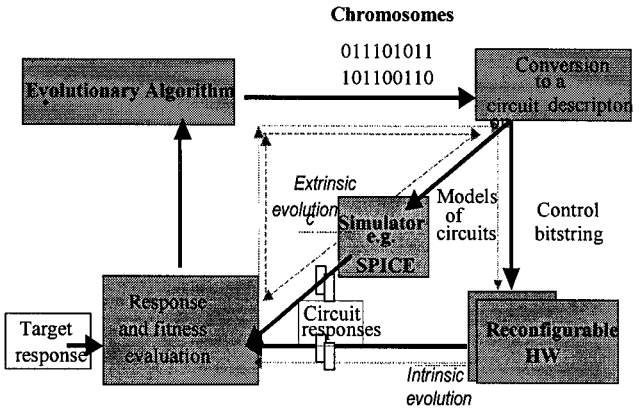

Figure 1 Evolutionary synthesis of electronic circuits

A solution determined by extrinsic evolution may eventually be downloaded or become blueprint for hardware. In intrinsic evolution the chromosomes are converted into control bitstrings, which are downloaded to program the reconfigurable device. The configuration bitstring determines the functionality of the cells of the programmable device and the interconnection pattern between cells. Circuit responses are compared against specifications of a target response and individuals are ranked based on how close they come to satisfying it. Preparation for a new iteration loop involves generation of a new population of individuals from the pool of the best individuals in the previous generation. Here, some individuals are taken as they were and some are modified by genetic operators, such as crossover and mutation. The process is repeated for a number of generations, resulting in increasingly better individuals. The process is usually ended after a given number of generations, or when the closeness to the target response has been reached. In practice, one or several solutions may be found among the individuals of the last generation.

3. Evolutionary Experiments

A variety of circuits have been synthesized through evolution. For example, Koza used Genetic Programming (GP) to grow an "embryonic" circuit to one that satisfies desired requirements [4]. This technique was used to evolve a variety of circuits, ranging from filters to controllers. Some of Koza's evolved designs rediscover solutions that at some point in time were patented, illustrating thus the power of the GP to obtain solutions that normally require an intelligent/innovative human. Some researchers succeded evolution in hardware. For example, evolution in hardware was demonstrated by Thompson [5], who used an FPGA as the programmable device, and a Genetic Algorithm (GA) as the evolutionary mechanism. More details on current work in evolvable hardware can be found in [6] and [7]. Evolutions of analog circuits reported in [4] were performed in simulations, without concern for a physical implementation, but rather as a proof-of-concept to show that evolution can lead to designs that compete with human designs, or even exceed them in performance.

Current programmable analog devices are very limited in capabilities and do not support the implementation of the designs resulted in simulations (but, in principle, one can test their validity in circuits built from discrete components, or in an ASIC). More recently, intrinsic evolutionary experiments were performed on commercial Field Programmable Analog Arrays (FPAA), custom-designed ASIC as well as other devices.

4. Building an evolvable system-on-a-chip

The efforts toward hardware evolution have been limited to simple circuits. In particular, for analog circuits, this limitation comes from a lack of appropriate reconfigurable analog devices to support the search, which precludes searches directly in hardware and requires evolving in software on hardware device models. Such models require evaluation with circuit simulators such as SPICE; the simulators need to solve differential equations and, for anything beyond simple circuits, they require too much time for practical searches of millions of circuit

solutions. A hardware implementation may offer a substantial advantage in circuit evaluation time; in certain cases the time for hardware evaluation can be seconds instead of days when evaluation is in software.

For efficiency of EHW applications, future reconfigurable devices would benefit from *implementing evolution-oriented* reconfigurable architectures (EORA). One of the most important features for EORA relates to the *granularity* of the programmable chip. FPAA offer only coarse granularity which is a clear limitation; FPGAs are offered both in versions with coarse grained and fine grained architectures (going to gate level as the lowest level of granularity). From the EHW perspective, it is interesting to have *programmable granularity*, allowing the sampling of novel architectures together with the possibility of implementing standard ones. The optimal choice of elementary block type and granularity is task dependent. At least for experimental work in EHW, it appears a good choice to build reconfigurable hardware based on elements of the lowest level of granularity. Virtual higher-level building blocks can be considered by imposing programming constraints. Ideally, the "virtual blocks" for evolution should be automatically defined/clustered during evolution. In addition EORA should be *transparent architectures*, allowing the analysis and simulation of the evolved circuits. They should also be robust enough not to be *damaged* by any configuration existent in the search space, potentially sampled by evolution. Finally, EORA should allow evolution of both analog and digital circuits.

An evolvable system-on-a-chip architecture is suggested in Figure 2. The main components are a Field Programmable Transistor Array and a Genetic Processor. The idea of a field programmable transistor array was introduced in [8] as a first step toward EORA. The FPTA is a concept design for hardware reconfigurable at transistor level. As both analog and digital CMOS circuits ultimately rely on functions implemented with transistors, the FPTA appears as a versatile platform for the synthesis of both analog and digital (and mixed-signal) circuits. The architecture is cellular, and has similarities with other cellular architectures as encountered in FPGAs (e.g. Xilinx X6200 family) or cellular neural networks. One key distinguishing characteristic relates to the definition of the elementary cell. The architecture is largely a "sea of transistors" with interconnections implemented by other transistors acting as signal passing devices (gray-level switches), and with islands of RC resources in between.

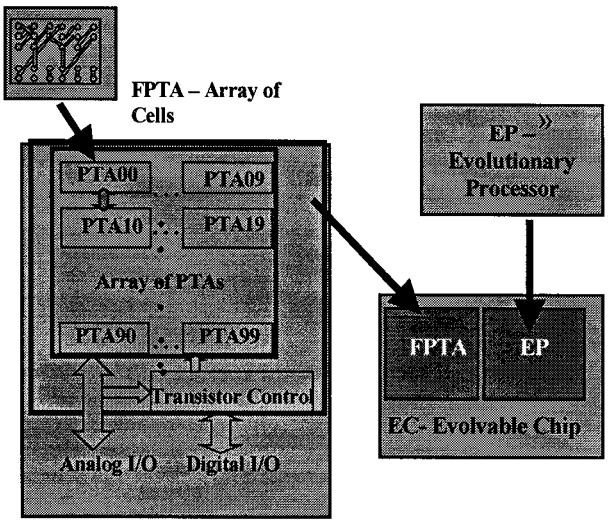

Figure 2 An evolvable SOC will integrate a Field Programmable Transistor Array amd and Evolutionary Processor

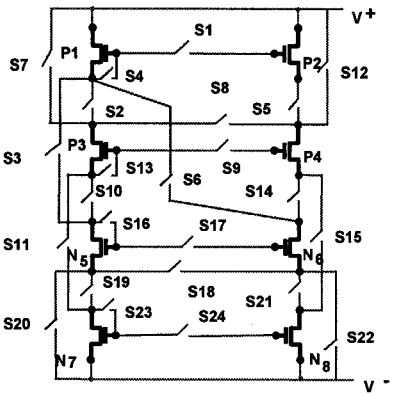

Figure 3. FPTA cell consisting of 8 transistors and 24 programmable switches.

The status of the switches (ON or OFF) determines a circuit topology and consequently a specific response. Thus, the topology can be considered as a function of switch states, and can be represented by a binary sequence, such as "1011...", where by convention one can assign 1 to a switch turned ON and 0 to a switch turned OFF. Programming the switches ON and OFF defines a circuit for which the effects of non-zero, finite impedance of the switches can be neglected in the first approximation (for low frequency circuits).

5. Evolving reconfigurable circuits for fuzzy logics

This section illustrates the evolutionary design of infinitesimal multi-valued logic circuits, more precisely circuits for fuzzy logics. The objective is to determine circuit implementations for conjunctions and disjunctions for fuzzy logics. In such logics, conjunction and disjunction are usually interpreted by a *T-norm* and by its dual *T-conorm* (*S-norm*) respectively. A function **T**: [0,1] x [0,1] => [0,1] is called a triangular norm (T-norm for short) if it satisfies the following conditions:

- associativity (**T**(x,**T**(y,z)) = **T**(**T**(x,y),z)),
- commutativity (**T**(x,y) = **T**(y,x)),
- monotonicity (**T**(x,y) ≤ **T**(x,z), whenever y ≤ z), and
- boundary condition (**T**(x,1) = x).

A function **S**: [0,1] x [0,1] => [0,1] is called a triangular conorm (T-conorm or S-norm for short) if it satisfies the conditions of associativity, commutativity, monotonicity, and the boundary condition **S**(x,0) = x. **S** and **T** are corresponding (or pairs) if they comply with De Morgan's laws. Frank's parametric T-norms and T-conorms (also refered to as fundamental T-norms/conorms in [9]) were the selected choice for modeling the logical connectives. The family of Frank T-norms is given by

$$T_s(x,y) = \begin{cases} MIN(x,y) & \text{if } (s = 0) \\ x \,.\, y & \text{if } (s = 1) \\ \log_s\left[1 + \dfrac{(s^x - 1).(s^y - 1)}{s - 1}\right] & \text{if } (0 < s < \infty),\ s \neq 1 \\ MAX(0, x + y - 1) & \text{if } (s = \infty) \end{cases} \quad (1)$$

The family of Frank T-conorms is given by

$$S_s(x,y) = \begin{cases} MAX(x,y) & \text{if } (s = 0) \\ x + y - x.y & \text{if } (s = 1) \\ 1 - \log_s\left[1 + \dfrac{(s^{1-x} - 1).(s^{1-y} - 1)}{s - 1}\right] & \text{if } ((0 < s < \infty),\ s \neq 1) \\ MIN(1, x + y) & \text{if } (s = \infty) \end{cases} \quad (2)$$

Electronic circuits implementing the above equations can be used in implementations of fuzzy logic computations or in implementing fuzzy S-T neurons. One interesting application made possible in this implementation is the selection of the most appropriate s-parameter for the application at hand. Examples of the influence of various T-norms and S-norms in fuzzy control and automated reasoning applications can be found in [10] and [11], and for learning in fuzzy neurons in [12].

The following preliminary results illustrate the possibility of evolving circuits that implement T and S for various values of the parameter *s*. The circuits were powered at 5V and the signal excursion was chosen between 1V (for logical level "0") and 4V (for logical level "1"). Intermediary values were in linear correspondence, i.e. 2.5V corresponds to logic level 0.5. etc. The experiments were performed both in software (Spice simulations) and in hardware using 2 FPTA cells. The experiments used a population size of 128 individuals, were performed for 400 generations (with uniform crossover, 70% crossover rate, 4% mutation rate, tournament selection) and took around 15 minutes using 16 processors when evolving in simulations. Each switch in the FPTA cell has a control bit associated with it in a direct mapping. Thus there are 24 bits in the chromosome describing one cell. Interconnections experiments were done mostly with 4 bits. Thus a 2 cell experiment would use 52bits (24*2+4).

Figures 4,5,6 show the response of circuits targeting the implementation of fundamental T-norms for s=0, s=1, and s=100 respectively. The diamond symbol (◊) marks points of simulated/measured response of evolved circuit, while the cross symbol (+) marks the points of an ideal/target response for the given inputs. The output (T) is mapped on the vertical axis; values on axis are in Volts. The circuit for T-norm with s=100 is shown mapped on two FPTA cells in Figure 7. Figure 8 shows the response of the circuit implementing the fundamental S-norm for s=100. Figure 9 shows the diagonal cut for the same S-norm. All these responses were for circuits evolved in software; for comparison the response of a circuit evolved in hardware (for s=100) is shown in Figure 10. Sometimes the actual response has a higher voltage value (◊ above +) than the ideal response for that input pair, sometimes is has a lower value (◊ below +). The errors are observed mainly at the domain extremes. The convergence toward solution can be seen in Figure 11, where a function of the error of best individual is plotted across the number of generations.

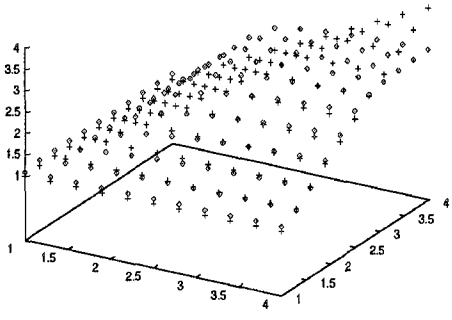

Figure 4 Simulated response of a circuit implementing the fundamental T-norm for s=0 (◊). Target characteristic shown with (+). x,y axis are for inputs, z (vertical) is the output, T. Axes are in Volts.

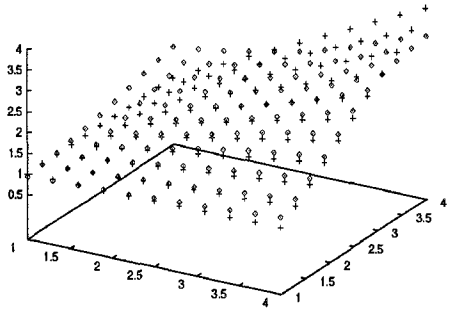

Figure 5 Response of a circuit implementing the fundamental T-norm for s=1 (◊). Target characteristic shown with (+).

The results presented here are a first and preliminary attempt at evolving fuzzy circuits. (*One should mention here the evolution of multivalued circuits reported in [13]. In [13] the search space is different, since the paper does not address the evolutionary synthesis of hardware functions, but the optimization of a network architecture, considering that the implementation of elementary functions is known. On a slightly different issue, one should remark here a distinction between the fuzzy case described here and the multivalued case: the numerical value of the output by operators described here is usually* not present *in the set of input values, making impossible a circuit solution based on switches selecting/routing input; e.g. for* $T_1(0.7,0.3)=0.21$, $0.21 \notin \{0.3, 0.7\}$). The purpose of the results presented in this paper is to illustrate what can be obtained in a rapid evolution, with no prior knowledge on the circuit solution, with no optimization in terms of Width and Length (W,L) of transistor channels, with limited resources (only those found in two FPTA cells). One limitation is the approximation error, ranging from 3.6% to a maximum of 9% MAPE (Mean Absolute Percent Error) in software and to a peak of 11.6% in hardware. Several factors can contribute to reducing the approximation error. One of them is to allow more flexibility in the selection of the points where the inputs are applied, and where the output is collected. In this experiment these were considered predetermined, however it is possible to let evolution decide where to interface the circuit with the input/output.

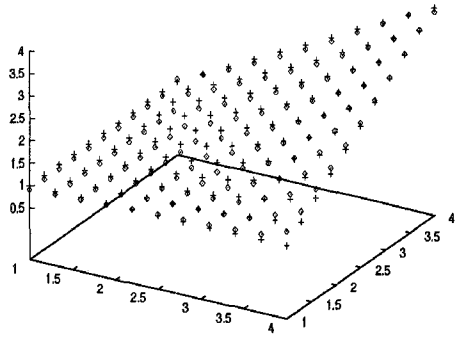

Figure 6 Response of a circuit implementing the fundamental T-norm for s=100 (◊). Target characteristic shown with (+).

Another way to increase the approximation power is to allow more resources, e.g. allow resources from more than 2 cells. This is similar to increasing the approximation power of neural networks when extra neurons are added. The described experiments do not have any parametric adjustment. The width and length of the transistor channel were considered fixed. However previous results indicate that parametric optimization can produce good adjustments after the topology has been determined [14]. This will also be possible in hardware since the new version of the chip

will allow switch-selectable transistors with different W/L in the same cell.

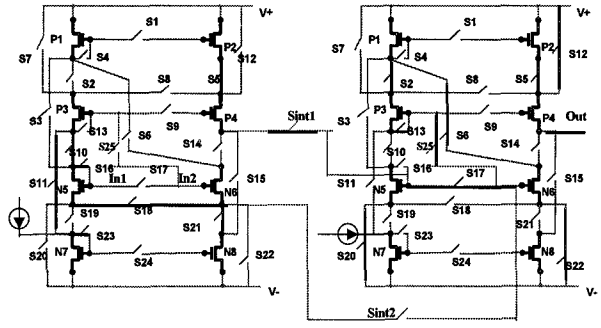

Figure 7 Evolved circuit implementing the fundamental T-norm for s=100 (with the response in Figure 8).

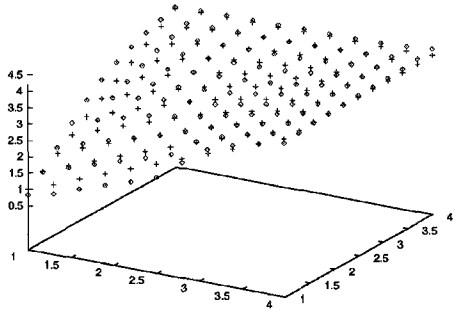

Figure 8 Response of a circuit implementing the fundamental S-norm for s=100 (◊). Target characteristic shown with (+).

These results are preliminary and are presented mainly to illustrate some aspects of the application of EHW to synthesis of electronic circuits implementing combinatorial fuzzy logic functions. No comparison with any state-of-the-art design tools is made, and, of course, the performance of (computer-assisted) human solutions could exceed the performance of the totally automated solutions illustrated here. However, to the author's best knowledge, complete automated design of the type presented here is not available in any other tool. Moreover, this author believes that completely automated techniques of the kind presented here will surpass current design techniques within the next 5-7 years. The role of the humans would shift toward providing specifications and evolutionary pressures to guide the design to the desired result (which is not a trivial task).

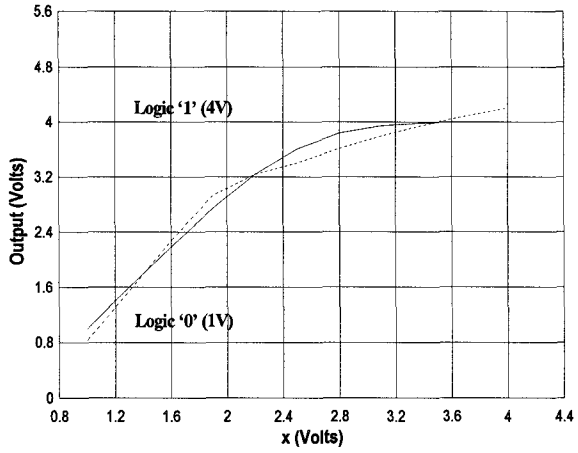

Figure 9 Diagonal cut for the response in Figure 10. Circuit implementing the fundamental S-norm for s=100. Target characteristic shown with full line.

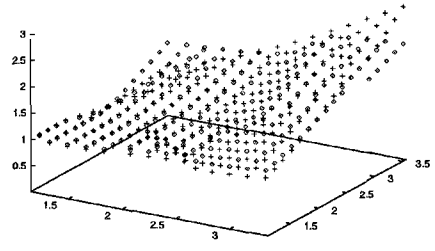

Figure 10 Measured response of a hardware-evolved circuit implementing the fundamental T-norm for s=100 (◊). Target characteristic shown with (+).

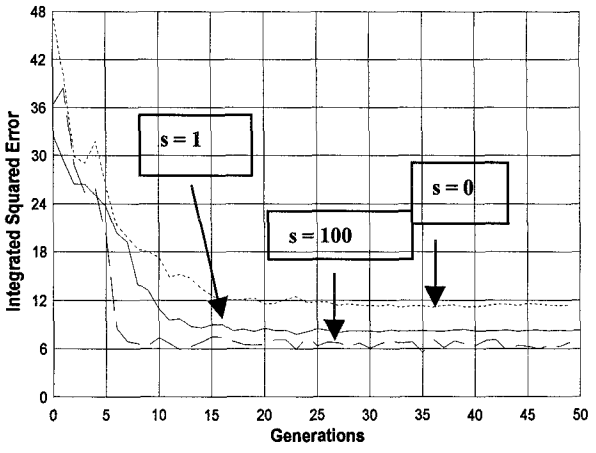

Figure 11 Decreasing error between best individual in each generation and target circuit, for the three software evolved circuits, with s=0, s=1, s=100.

7. Conclusion

This paper presented an effort toward building evolution-oriented devices and demonstrated how electronic circuits can be automatically synthesized, on-the-chip, to produce a desired functionality. It illustrated the aspects of using evolvable hardware for the design of unconventional circuits such as combinatorial circuits for fuzzy logics.

Acknowledgements

The work described in this paper was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the National Aeronautics and Space Administration.

References

[1] A. Stoica, D. Keymeulen, R. Tawel, C. Lazaro and Wei-te Li. "Evolutionary experiments with a fine-grained reconfigurable architecture for analog and digital CMOS circuits." In [7], 77-84

[2] R. Zebulum, A. Stoica and D. Keymeulen, "A Flexible Model of a CMOS Field Programmable Transistor Array Targeted for Hardware Evolution", Third Int. Conference on Evolvable Systems: From Biology to Hardware (ICES2000), Edinburgh, April 17-19, 2000, to appear

[3] A. Stoica, R. Zebulum and D. Keymeulen. Mixtrinsic Evolution. In T. Fogarty, J. Miller, A. Thompson and P. Thompson, (eds.), Proceedings of the Third International Conference on Evolvable systems: From Biology to Hardware (ICES2000), April 17-19, 2000, Edinburgh, UK. New York, USA, Springer Verlag, 208-217

[4] J. Koza, F.H. Bennett, D. Andre, and M.A Keane, "Automated WYWIWYG design of both the topology and component values of analog electrical circuits using genetic programming", *Proceedings of Genetic Programming Conference*, Stanford, CA, 1996, 28-31

[5] A. Thompson, "An evolved circuit, intrinsic in silicon, entwined in physics". In *International Conference on Evolvable Systems*. Springer-Verlag Lecture Notes in Computer Science, 1996, 390-405

[6] M. Sipper, D. Mange, A. Perez-Uribe (Eds.) Evolvable Systems: From Biology To Hardware, *Proc. of the Second International Conference, ICES 98*, Lausanne, Switzerland, Springer-Verlag Lecture Notes in Computer Science, 1998.

[7] A. Stoica, D. Keymeulen and J. Lohn (Eds.) *Proc. of the First NASA/DoD Workshop on Evolvable Hardware*, July 19-21, 1999, Pasadena, CA IEEE Computer Society Press.

[8] Stoica, A. "Toward evolvable hardware chips: experiments with a programmable transistor array." Proceedings of 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, Granada, Spain, April 7-9, IEEE Comp Sci. Press, 1999, 156-162

[9] Butnariu, D. and Klement, E. P., Triangular norm-based measures and games with fuzzy coalitions, Kluwer Academics, 1993

[10] M. M. Gupta and J. Qi, "Design of fuzzy logic controllers based on generalized t-operators", Fuzzy Sets and Systems, Vol. 40, 1991, 473-389

[11] M.M. Gupta and J. Qi, "Theory of T-norms and fuzzy inference methods", Fuzzy Sets and Systems, Vol. 40, 1991, 431-450

[12] Stoica, A., "Synaptic and somatic operators for fuzzy neurons: which T-norms to choose?" In Proc. of 1996 Biennial Conference of the North American Fuzzy Information Processing Society - NAFIPS, Berkeley, CA, June 19-22, 55-58

[13] W. Wang and C. Moraga, "Design of Multivalued Circuits using Genetic Algorithms", Proceedings 26th International Symposium on Multiple-Valued Logic, 1996, 216 -221

[14] Stoica, A. "On hardware evolvability and levels of granularity", Proc. of the International Conference "Intelligent Systems and Semiotics 97: A Learning Perspective, NIST, Gaithersburg, MD, Sept. 22-25, 1997

[15] A. Stoica, D. Keymeulen, V. Duong and C. Lazaro, "Automatic Synthesis and Fault-Tolerant Experiments on an Evolvable Hardware Platform", In R. Profet et al.(eds.), Proc. of IEEE Aerospace Conf., March 18-25, 2000, Big Sky, MO, IEEE Press (CD, paper # 7_1102)